

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR UNITED STATES PATENT

FOR

**METHOD AND APPARATUS FOR PRODUCING
A MULTICAST TREE**

Inventor:

Haixiang He
10 Westgate Drive #T3
Woburn, MA 01801

Attorney Docket: 2494/105
(13774BA)

Attorneys:

BROMBERG & SUNSTEIN LLP
125 Summer Street
Boston, MA 02110
(617) 443-9292

**METHOD AND APPARATUS FOR PRODUCING
A MULTICAST TREE
PRIORITY**

5 This patent application claims priority from provisional U.S. patent
application number 60/249,303, entitled, "METHOD FOR DETERMINING
MULTICAST DISTRIBUTION TREE," filed on November 16, 2000 and naming
Haixiang He as inventor, the disclosure of which is incorporated herein, in its
entirety, by reference.

10 **FIELD OF THE INVENTION**

The invention relates generally to data transmission networks and, more
particularly, the invention relates multicast transmissions within data
transmission networks.

15 **BACKGROUND OF THE INVENTION**

Multicasting is a well known method of transmitting messages to selected
groups of users across a network, such as the Internet. One simple example of
multicasting involves transmitting an E-mail message to a plurality of users that
20 each are on a mailing list. Video conferencing and teleconferencing also use
multicasting principles and, consequently, often are referred to as
"multiconferencing."

To enable users of a computer network to establish a multicast, the
network devices (e.g., routers or personal computers) associated with each user
25 of a multicast must be logically and/or physically connected. To that end, users
in a multicast are logically connected through a structure known in the art as a
"multicasting tree" (also referred to as a "multicast tree" or "tree structure").
Network management and multicasting applications use the multicast tree for a

number of purposes, such as for routing data in the multicast and troubleshooting multicast problems.

A number of different multicasting protocols have been developed to implement multicasting within a network. Among others, those protocols include the Protocol Independent Multicast ("PIM") and the Distance Vector Multicast Routing protocol ("DVMRP"). Like other such protocols, both PIM and DVMRP each build a multicast tree in accordance with processes that are defined specifically for such protocols. Information about such trees is stored in databases that are accessible to those specific protocols only. Accordingly, by way of example, a database with tree information for PIM cannot be read by an application used for DVMRP.

There are times, however, when applications that are configured for use with one multicast protocol must read tree information stored in a multicast database that is configured in another format. In such case, the multicast database cannot be read and thus, no such tree information can be recovered.

SUMMARY OF THE INVENTION

In accordance with one aspect of the invention, a method and apparatus for producing a multicast tree in a network traces such tree based upon information retrieved from a multicast database that is protocol independent. More particularly, the network includes a plurality of network devices. A set of the plurality of network devices each includes the prior noted protocol independent multicast database. After the multicast database in each of the set of network devices is located, the multicast information in such databases is retrieved. After being retrieved, the multicast information is traced across the plurality of network devices to form the multicast tree.

In some embodiments, the multicast tree includes a root node, and the retrieved information is traced from the root node. In such case, the root node is

one of the plurality of network devices. The network may implement the Internet Protocol, while each multicast database may be a management information base. In addition, the set of network devices may include a router, and the retrieved multicast information may be traced by an application
5 incorporating the Simple Network Management Protocol ("SNMP"). The set of network devices may include no more than one of the plurality of network devices, or may include more than one of the plurality of network device.

Each of the set of network devices may include another type of database that is protocol independent. For example, the set of network devices may
10 include a protocol independent unicast database having network information. The multicast tree may be produced also based upon the network information in each unicast database. In particular, after locating the unicast database within each of the set of network devices, the network data within such databases is retrieved. Once retrieved, the network data is traced to form the multicast tree.

15 Various embodiments of the invention are implemented as a computer program product having a computer usable medium with computer readable program code thereon. The computer readable code may be read and utilized by a computer system in accordance with conventional processes.

20 BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing description of various embodiments of the invention should be appreciated more fully from the following further description thereof with reference to the accompanying drawings wherein:

Figure 1 schematically shows a well known network arrangement that
25 may be used in connection with illustrative embodiments of the invention.

Figure 2 generally shows a process of producing a multicast tree in accordance with illustrative embodiments of the invention.

Figure 3 schematically shows nodes in a simplified multicast for the purpose of illustrating the process in figure 2.

Figure 4 shows a more detailed example of forming a multicast tree in accordance with illustrative embodiments of the invention.

5

DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

In illustrative embodiments of the invention, a multicast tree is formed based upon information retrieved from a multicast database that is protocol independent. By way of example, the database may be a management
10 information base ("MIB"). Details are discussed below.

Figure 1 schematically shows a network 10 that may be used with illustrative embodiments of the invention. The network 10 includes a plurality of network devices that communicate in accordance with various known data transmission protocols. For example, the network devices may communicate
15 with remote network devices (not shown) across the Internet by using the Transport Control Protocol and the Internet Protocol ("TCP/IP"). Each shown network device may communicate with other shown network devices by using link layer protocols, such as the Frame Relay protocol.

The network 10 includes a plurality of network devices, including routers
20 12, personal computers 14, servers 16, external data storage devices 18, and network appliances 20 (*e.g.*, an Internet telephone). Of course, other types of network devices not shown can also be included in the network 10, such as switches, hubs, multiplexers, etc Accordingly, details of the shown network 10 are intended to be exemplary and thus, various embodiments are not limited
25 to the specific configuration of the shown network 10.

The network 10 includes five interconnected routers 12. Each router is referred to herein as one of router 1, router 2, router 3, router 4, or router 5. When discussed generally, however, the routers are identified herein by

reference number "12." designated as and individually referred to as one of routers 1-5. The routers 12 may communicate and be managed by applications implementing any known routing protocol, such as the Simple Network Management Protocol ("SNMP") or the Routing Information Protocol ("RIP").

- 5 For example, router 1 may use SNMP to update the other routers in the network 10 when it detects a path change or other change in the network 10.

Routers 1, 2, 4, and 5 each are a part of separate local area networks ("LANs") that include the other types of network devices. In particular, the LAN of router 1 includes a connection to the Internet (e.g., via an Internet service
10 provider), a personal computer 14, a network appliance 20, a server 16, and an external data storage device 18. Such LAN communicates directly with the LAN of router 2 via a direct link between the respective routers. In a similar manner, the LAN of router 4 is directly connected to the LAN of router 5 via a direct link between such two routers. Router 3 connects the LANs of router 2 and router 4.

15 Routers 1-4 each have an associated multicast database that is protocol independent. Such multicast database includes multicast data relating to one of the four routers. More particularly, for a given router, each such database includes multicast data, such as the group multicast ID of one or more multicasts, the IP address of parent and child nodes to the given router for the
20 one or more multicasts, and the interfaces of such parent and child nodes. Accordingly, the multicast database on any node does not include enough information to build the multicast tree.

The multicast database may be stored on each node as a MIB. Each such MIB on any given node is shown and referred to herein as the "standard MIB
25 22." Router 1 thus has access to its standard MIB 22 via a coupled external storage device 18 (e.g., a disk array), while routers 2-4 each have internal memory for storing their standard MIBs 22. As discussed in greater detail below, the standard MIB 22 on each node in a multicast is used to form a multicast tree. It

also should be noted that since router 5 does not include its own standard MIB 22, it cannot implement illustrative embodiments of the invention.

In addition to the standard MIB 22, each of routers 1-4 also includes a unicast database 24 normally used for routing data packets across the network
5 10. As known by those in the art, the unicast database 24 also is protocol independent. Accordingly, the routers can use conventionally known program code to access their respective unicast databases 24. In some embodiments, such conventionally known program code also can be used to access the standard MIB 22.

10 A subset of the network devices in the network 10 together form and maintain a multicast in accordance with conventional processes. For example, such network devices may form and maintain a multicast with one of several known multicast protocols, such as the Protocol Independent Multicast ("PIM") or the Distance Vector Multicast Routing protocol ("DVMRP"). Of course, any
15 known multicast protocol may be used and thus, embodiments of the invention are not limited to the discussed protocols.

The standard MIB 22 on each node preferably is formed and maintained as the multicast is formed and maintained. In addition, as noted above, the standard MIBs 22 on each node may include multicast data for a plurality of
20 multicasts and thus, be used to trace a plurality of multicast trees. The standard MIB 22 may be similar to that described in Request For Comment number 2932 ("RFC 2932"), which is published by the Internet Engineering Task Force. The disclosure of RFC 2932 is incorporated herein, in its entirety, by reference. The purpose of the standard MIB 22 described in RFC 2932, as understood by the
25 inventor, is to store state information and not to build the multicast tree. It also should be noted that illustrative embodiments may be directed to other types of multicast databases that are protocol independent. Accordingly, the standard

MIB 22 is discussed herein as but one exemplary type of database and thus, not intended to limit the scope of various embodiments of the invention.

More particularly, each standard MIB 22 includes managed objects for managing IP multicast routing for IPv4. In illustrative embodiments, the standard MIB 22 includes one scalar and five tables. The tables include:

1. Multicast Route Table having multicast routing information for IP datagrams sent by particular sources to the IP multicast groups known to a router.

2. IP Multicast Routing Next Hop Table having information on the next hops of the routing of IP multicast datagrams. Each entry is one of a list of next hops on outgoing interfaces for particular sources sending to a particular multicast group address.

3. IP Multicast Routing Interface Table having multicast routing information specific to interfaces.

4. IP Multicast Scope Boundary Table having the boundaries configured for multicast scopes. For additional details on this table, see Meyer, D., "Administratively Scoped IP Multicast," BCP 23, RFC 2365, July 1998, the disclosure of which is incorporated herein, in its entirety, by reference.

5. IP Multicast Scope Name Table having human readable names of multicast scope.

As noted above, the standard MIB 22 can be used by a network management application, such as one implementing SNMP, to build a multicast

tree representing the multicast. To that end, figure 2 generally shows an illustrative process of forming the multicast tree for a given multicast. The process begins at step 200, in which the root node of the multicast is located. The standard MIB 22 for that node then is accessed to determine the multicast data relating to such root node (step 202). Among other things, such multicast data can include the IP addresses of the next hop(s) children nodes, interfaces of the such child nodes, etc Information relating to the root node then is stored in a tree data structure that is formed to ultimately contain the entire tree (step 204). Such information may include the IP address of the root node, and pointers child nodes. The data structure may be stored in a memory device that is resident on the network device executing the SNMP management program.

After the multicast data is stored in the tree data structure, it is determined at step 206 if additional nodes are in the multicast. If not, then the entire multicast has been traced, thus completely forming the multicast tree. Since at this point the root node is being examined, then other nodes do exist. Accordingly, the process continues to step 208, in which the standard MIB 22 in the first child node of the root node is queried. The process then loops back to step 202, in which multicast data is retrieved.

This process repeats for all child nodes of the root node. After the last child node for the root node is processed, then the process repeats for each child node of the root node. In such case, the child nodes to each child of the root node (i.e., the grandchild nodes of the root node) are ascertained and stored in the tree data structure. This iterative process continues until all nodes and all branches in the multicast have been traced, thus producing the multicast tree within the tree data structure. The final resulting tree data structure may be used for a plurality of reasons, such as by a network administrator or application program for troubleshooting purposes.

Below is pseudo code for an exemplary implementation of illustrative embodiments that have been implemented using JAVA. As explicitly noted below with reference to figure 4, the multicast tree is formed by indexing to the IP address of the source of the multicast traffic, and the multicast group address.

5 The pseudo code is as follows:

Create a queue of tree nodes;

Add the root node into queue;

10 While (Queue is not empty) {

 Get and delete the first tree node element **TN** from queue;

 Query the **ipMRouteNextHopEntry** table of the multicast routing mib **Mcast MIB** using SNMP with the SNMP target set to **TN**;

15 Get all the outgoing network interface indexes with the conditions that the interfaces are used to forward multicast traffic from source IP address **S** and destination group address **G**;

 For each of the outgoing interface index **OI** shown above {

20 Query the topology information with the tree node and the interface index to get all the neighbors of this interface;

 For each of the neighbors **NB** {

25 If the neighbor is a host receiver, then add it as a child node of **TN**;

 If the neighbor is a router, then {

30 Query
 ipMRouteEntry.ipMRouteInIfIndex of **Mcast MIB** using SNMP with the SNMP target set to the neighbor;

 Get the incoming network interface index **II** with the condition that **ipMRouterEntry's** instance **ID**

10

*contains both source IP address S
and destination group address G;*

*If there is a link that contains TN,
OI, NB, II, then {*

*Add the neighbor as a child node of
TN;*

Add the neighbor into the queue;

}

}

}

}

}

Figure 3 shows a simplified group of network devices implementing a multicast. This group of network devices will be used by way of example to illustrate one implementation of the pseudo code and figure 2 in accordance with various embodiments of the invention. The multicast includes a source network device (e.g., a personal computer 14), a first router designated as R1, a second router designated as R2, a first host (e.g., a personal computer 14) and a second host (e.g., a personal computer 14). The network devices in this network may be a part of the network 10 shown in figure 1, or any other network that implements illustrative embodiments. Each network device in the group has one or more interfaces for connecting with a neighboring network device. The interfaces are designated in figure 3 by the numbers 1, 2, and 3.

Figure 4 shows a specific process using SNMP for forming the multicast tree for the group of network devices shown in figure 3. In a manner similar to the process shown in figure 2, this process traces the multicast tree from node to node in the multicast. When discussing this process, the source IP address is

10.100.50.2, and the multicast destination group address is 224.3.3.3.

Accordingly, as demonstrated below, these are the two above noted parameters that are used to form the multicast tree.

The process begins at step 400, in which a node queue and a tree data structure (discussed above) are formed. As discussed below, the queue stores list of IP addresses of each node in the multicast as they is learned. This list then is used by the process to step through the entire multicast tree to ensure that no nodes are omitted from the final tree structure. The tree data structure, as noted with reference to figure 2, is used to build the overall multicast tree structure.

Shortly after initialization, the source S1 and its router R1 are added to the tree data structure.

Once the queue and tree structure are initialized, the process continues to step 402, in which the root node (router R1) is added to the queue. A pointer in the queue then points to router R1, which causes the management application to access the MIB data in the router R1 to determine additional information relating to the multicast tree (step 404). Specifically, the IP Multicast Next Hop table of the router R1 is accessed to determine the next hop in the multicast from router R1. This table is referred to in the pseudo code as the ipMRouteNextHopEntry" table. The source address (10.100.50.2) and the multicast group destination address (224.3.3.3) are used to determine the interfaces of the router R1 that have connected network devices in the multicast. Continuing with the exemplary network in figure 3, it is determined that interface 1 of router R1 is connected to one such network device, while interface number 2 of router R1 is connected to another such network device.

The process then continues to step 408, in which all child nodes of the router R1 are added to the queue. Accordingly, the IP addresses of the router R2 and host H1 are added to the queue. Moreover, the IP addresses of the child

nodes are added to the tree data structure with pointers indicating that such nodes are child nodes of router R1. In some embodiments, interface information also is added to the tree data structure.

The process then continues to step 412, in which it is determined if more nodes are in the multicast tree. To that end, the pointer in the queue is incremented to point to the next node, which is either one of router R2 or host H1. In such case, the standard MIB 22 in such next node is examined to determine next hop information to it (step 404). This iterative process thus continues until the entire multicast tree is formed in the tree data structure. This may be determined when a null or other value is pointed to in the queue at step 412.

In alternative embodiments, in addition to the data in the standard MIB 22, data from the unicast database 24 also may be used to build the multicast tree. For example, both the standard MIB 22 and the unicast database 24 may be queried to ascertain multicast tree information. Of course, in this embodiment, the multicast tree still is formed without accessing protocol specific multicast databases.

Accordingly, the entire multicast tree can be traced and thus, fully formed, by accessing the standard MIB 22. Since it is not dependent on any specific multicast routing protocol, the process of forming the multicast tree can be used by network systems that support different multicast routing protocols.

Some embodiments of the invention may be implemented at least in part in any conventional computer programming language comprising computer program code. For example, preferred embodiments may be implemented in a procedural programming language (*e.g.*, "C") or an object oriented programming language (*e.g.*, "C++"). Alternative embodiments of the invention may be implemented, at least in part, as preprogrammed hardware elements (*e.g.*,

application specific integrated circuits, FPGAs, and digital signal processors), analog circuit elements, or other related components.

In other embodiments, the disclosed apparatus and method may be implemented as a computer program product for use with a computer system.

5 Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable medium (*e.g.*, a diskette, CD-ROM, ROM, or fixed disk) or transmittable to a computer system, via a modem or other interface device, such as a communications adapter connected to a network over a medium. The medium may be either a tangible medium (*e.g.*,
10 optical or analog communications lines) or a medium implemented with wireless techniques (*e.g.*, microwave, infrared or other transmission techniques). The series of computer instructions embodies all or part of the functionality previously described herein with respect to the system. Those skilled in the art should appreciate that such computer instructions can be written in a number of
15 programming languages for use with many computer architectures or operating systems. Furthermore, such instructions may be stored in any memory device, such as semiconductor, magnetic, optical or other memory devices, and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies. It is expected that such a
20 computer program product may be distributed as a removable medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the network (*e.g.*, the Internet or World Wide Web). Of course, some embodiments of the
25 invention may be implemented as a combination of both software (*e.g.*, a computer program product) and hardware. Still other embodiments of the invention are implemented as entirely hardware, or entirely software (*e.g.*, a computer program product).

Although various exemplary embodiments of the invention have been disclosed, it should be apparent to those skilled in the art that various changes and modifications can be made that will achieve some of the advantages of the invention without departing from the true scope of the invention. These and
5 other obvious modifications are intended to be covered by the appended claims.

continued on next page